

**CLEAN VERSION OF REPLACEMENT PARAGRAPHS**

Please enter the following replacement paragraphs to the specification.

**1. Please amend the paragraph at page 10, line 21- page 11, line 8, as follows:**

The computer 20 may operate in a networked environment using logical connections to one or more remote computers, such as remote computer 49. These logical connections are achieved by a communication device coupled to or a part of the computer 20; the invention is not limited to a particular type of communications device. The remote computer 49 may be another computer, a server, a router, a network PC, a client, a peer device or other common network node, and typically includes many or all of the elements described above relative to the computer 20, although only a memory storage device 50 has been illustrated in FIG. 1. The logical connections depicted in FIG. 1 include a local-area network (LAN) 51 and a wide-area network (WAN) 52. Such networking environments are commonplace in office networks, enterprise-wide computer networks, intranets and the Internet, which are all types of networks.

**2. Please amend the paragraph at page 11, lines 9-19, as follows:**

When used in a LAN-networking environment, the computer 20 is connected to the local network 51 through a network interface or adapter 53, which is one type of communications device. When used in a WAN-networking environment, the computer 20 typically includes a modem 54, a type of communications device, or any other type of communications device for establishing communications over the wide area network 52, such as the Internet. The modem 54, which may be internal or external, is connected to the system bus 23 via the serial port interface 46. In a networked environment, program modules depicted relative to the personal computer 20, or portions thereof, may be stored in the remote memory storage device. It is appreciated that the network connections shown are exemplary and other means of and communications devices for establishing a communications link between the computers may be used.

**3. Please amend the table at page 15, between lines 14 and 15, as follows:**

(1) $fn(\mathbf{0}) \triangleq \phi$	(8) $fn(n) \triangleq \{n\}$
(2) $fn(P Q) \triangleq fn(P) \cup fn(Q)$	(9) $fn(in M) \triangleq fn(M)$
(3) $fn(!P) \triangleq fn(P)$	(10) $fn(out M) \triangleq fn(M)$
(4) $fn(M[P]) \triangleq fn(M) \cup fn(P)$	(11) $fn(open M) \triangleq fn(M)$
(5) $fn(M.P) \triangleq fn(M) \cup fn(P)$	(12) $fn(\epsilon) \triangleq \phi$
(6) $fn((n).P) \triangleq fn(P) - \{n\}$	(13) $fn(M.M') \triangleq fn(M) \cup fn(M')$
(7) $fn(\langle M \rangle) \triangleq fn(M)$	

**4. Please amend the paragraph at page 15, line 16 – page 16, line 8 as follows:**

The thirteen statements within this table are explained as follows. The first statement states that there are no free names for the inactivity process. The symbol  $\triangleq$  specifies that the left-hand side of the symbol is defined as the right-hand side of the symbol. This definition is applicable in any statement in which the symbol  $\triangleq$  appears. The second statement states that the free names for the composition  $P|Q$  are the free names for  $P$  conjoined with the free names for  $Q$ . The third statement states that when a process is replicated from another process, it has the same free names as that latter process. The fourth statement states that the free names of a container  $M$  having therein a process  $P$  are the free names of  $M$  by itself conjoined with the free names of  $P$  – that is,  $M[P]$  cannot take on any names that are not allowed by either  $M$  itself or  $P$  itself. The fifth statement states that the free names of the capability action  $M.P$  cannot take on any names that are not allowed by either  $M$  itself or  $P$  itself. The sixth statement states that the free names of the input action  $(n).P$  are the free names of the process  $P$ , minus the name  $n$ .

**5. Please amend the paragraph at page 18, line 29- page19, line 11, as follows:**

Finally, the following syntactic conventions and abbreviations, as summarized in the next table, are used herein. A fact is also provided.

Syntactic conventions

$!P Q$	is read	$(!P) Q$
$M.P Q$	is read	$(M.P) Q$
$(n).P Q$	is read	$((n).P) Q$

Abbreviations

$n[ ]$	$\triangleq n[0]$
$M$	$\triangleq M.0$ (where appropriate)

Fact

$$n[P] \equiv m[P'] \text{ iff } n = m \text{ and } P \equiv P''$$

**6. Please amend the table at page 19, following line 20, as follows:**

$A, B, C ::=$

1	$T$	true
2	$\neg A$	negation
3	$A \vee B$	disjunction
4	$n[A]$	location
5	$A' A''$	composition
6	$\exists n.A$	existential quantification over names
7	$\diamond A$	somewhere modality (spatial)
8	$\lozenge A$	sometime modality (temporal)
9	$A@n$	location adjunct
10	$A>B$	composition adjunct

**7. Please amend the paragraph at page 20, lines 1-12 as follows:**

The logical formulas of the preceding table are described as follows. The first statement is a logical true, while the second statement is a logical negation and the third statement is a logical disjunction. The fourth statement means that the process A is located within the container, or ambient, n. The fifth statement is a logical composition. The sixth statement specifies the existential quantifier operation, that there is some process A within the container named n. The seventh statement specifies a spatial operator, that somewhere, at some location, the process A exists. That is, within some container, anywhere in the domain space being considered, the process A exists. Similarly, the eighth statement specifies a temporal operator, that at some point in time, the process A will exist (or currently exists). The ninth statement specifies that the process A exists within the container named n. Finally, the tenth statement is a logical composition adjunct.

**8. Please amend the table on page 20, following line 12, as follows:**

1 F	$\triangleq \neg T$	false
2 A $\wedge$ B	$\triangleq \neg(\neg A \vee \neg B)$	conjunction
3 A $\Rightarrow$ B	$\triangleq \neg A \vee B$	implication
4 A $\Leftrightarrow$ B	$\triangleq (A \Rightarrow B) \wedge (B \Rightarrow A)$	logical equivalence
5 A $\sqcup$ B	$\triangleq \neg(\neg A \mid \neg B)$	decomposition
6 !A	$\triangleq A \sqcup F$	every component satisfies A
7 ?A	$\triangleq A \mid T (\Leftrightarrow \neg !\neg A)$	some component satisfies A
8 $\forall n. A$	$\triangleq \neg \exists n. \neg A$	universal quantification over names
9 $\Box A$	$\triangleq \neg \Diamond \neg A$	everywhere modality (spatial)
10 $\Box A$	$\triangleq \neg \Diamond \neg A$	everytime modality (temporal)
11 A @	$\triangleq \forall n. A @n$	in every location context
12 $\triangleright A$	$\triangleq T \triangleright A$	in every composition context

**9. Please amend the paragraph at page 20, line 16- page 21, line 12 as follows:**

The derived connectives of the preceding table are explained as follows. The first statement is the logical false, and is derived and defined as a function of the logical true. The second statement is the logical conjunction, while the third statement is the logical implication and the fourth logical equivalence. The fifth statement specifies logical decomposition. The sixth statement defines  $\forall A$  as universal satisfaction, that every component satisfies the process  $A$ . Likewise, the seventh statement defines  $\exists A$  as partial satisfaction, that some component satisfies the process  $A$ . The eighth statement defines the universal quantifier  $\forall$  in terms of the existential quantifier  $\exists$ ; that all the processes  $A$  are within the container  $n$ . The ninth statement states that the process  $A$  exists everywhere, from a spatial perspective, while the tenth statement states that the process  $A$  has existed, and still exists, at every time. The eleventh and twelfth statements specify the in every location context and the in every composition context, respectively, and are derived from the ninth and tenth logical formula statements of the logical formulas table.

**10. Please amend the paragraph at page 21, lines 16-17, as follows:**

*A/10*

- Infix ' $\triangleright$ ' binds stronger than '|', and they both bind stronger than the standard logical connectives.

**11. Please amend the paragraph at page 21, line 22 – page 22, line 2 as follows:**

The satisfaction relation  $P \models A$  (process  $P$  satisfies formula  $A$ ) is defined inductively in the following tables, where  $\Pi$  is the sort of processes,  $\Phi$  is the sort of formulas, and  $\Lambda$  is the sort of names. Quantification and sorting of meta-variables are made explicit because of subtle scoping issues, particularly in the definition of  $P \models \exists n.A$ . Similar syntax for logical connectives is used at the meta-level and object-level.

**12. Please amend the table on page 22, between line 11 and line 13, as follows:**

$\forall P:\Pi.$	$P \models T$	$\triangleq$	
$\forall P:\Pi,A:\Phi$	$P \models \neg A$	$\triangleq$	$\neg P \models A$
$\forall P:\Pi,A,B:\Phi.$	$P \models A \vee B$	$\triangleq$	$P \models A \vee P \models B$
$\forall P:\Pi,n:\Lambda,A:\Phi.$	$P \models n[A]$	$\triangleq$	$\exists P':\Pi. P \equiv n[P'] \wedge P' \models A$
$\forall P:\Pi,A,B:\Phi.$	$P \models A \mid B$	$\triangleq$	$\exists P',P'':\Pi. P \equiv P' \mid P'' \wedge P' \models A \wedge P'' \models B$
$\forall P:\Pi,n:\Lambda,A:\Phi.$	$P \models \exists n.A$	$\triangleq$	$\exists m:\Lambda. P \models A\{n \leftarrow m\}$
$\forall P:\Pi,A:\Phi$	$P \models \diamond A$	$\triangleq$	$\exists P':\Pi. P \downarrow^* P' \wedge P' \models A$
$\forall P:\Pi,A:\Phi$	$P \models \lozenge A$	$\triangleq$	$\exists P':\Pi. P \rightarrow^* P' \wedge P' \models A$
$\forall P:\Pi,A:\Phi$	$P \models A @ n$	$\triangleq$	$n[P] \models A$
$\forall P:\Pi,A,B:\Phi.$	$P \models A \triangleright B$	$\triangleq$	$\forall P':\Pi. P' \models A \Rightarrow P   P' \models B$

**13. Please amend the paragraph on page 22, lines 19-20 as follows:**

A13 • A process  $P$  satisfies the  $n[A]$  formula if there exists a process  $P'$  such that  $P \equiv n[P']$  and  $P' \models A$ .

**14. Please amend the paragraph at page 22, lines 21-22 as follows:**

A14 • A process  $P$  satisfies the  $A \mid B$  formula if there exist processes  $P'$  and  $P''$  such that  $P \equiv P' \mid P''$  with  $P'$  satisfying  $A$  and  $P''$  satisfying  $B$ .

**15. Please amend the paragraph at page 23, lines 6-7 as follows:**

A15 • A process  $P$  satisfies the formula  $\lozenge A$  if  $A$  holds in the future for some residual  $P'$  of  $P$ , where "residual" is defined by  $P \rightarrow^* P'$ .

**16. Please amend the paragraph at page 23, lines 10-14, as follows:**

A16 • A process  $P$  satisfies the formula  $A \triangleright B$  if, given any parallel context  $P'$  satisfying  $A$ , the combination  $P | P'$  satisfies  $B$ . Another reading of  $P \models A \triangleright B$  is that  $P$  manages to satisfy  $B$  under any possible attack by an opponent that is bound to satisfy  $A$ . Moreover, " $P$  satisfies  $(\Box A) \triangleright (\Box B)$ " means that  $P$  preserves the invariant  $A$ .

**17. Please amend the table at page 23, between lines 14 and 16, as follows:**

$\forall P:\Pi.$	$\neg P \models F$
$\forall P:\Pi, A, B: \Phi.$	$P \models A \wedge B$ iff $P \models A \wedge P \models B$
$\forall P:\Pi, A, B: \Phi.$	$P \models A \Rightarrow B$ iff $P \models A \Rightarrow P \models B$
$\forall P:\Pi, A, B: \Phi.$	$P \models A \Leftrightarrow B$ iff $P \models A \Leftrightarrow P \models B$
$\forall P:\Pi, A, B: \Phi.$	$P \models A \sqcup B$ iff $\forall P', P'': \Pi. P \equiv P' P'' \Rightarrow P' \models A \vee P'' \models B$
$\forall P:\Pi, A: \Phi.$	$P \models !A$ iff $\forall P', P'': \Pi. P \equiv P' P'' \Rightarrow P' \models A$
$\forall P:\Pi, A: \Phi.$	$P \models ?A$ iff $\exists P', P'': \Pi. P \equiv P' P'' \wedge P' \models A$
$\forall P:\Pi, n: \Lambda, A: \Phi.$	$P \models \forall n.A$ iff $\forall m: \Lambda. P \models A\{n \leftarrow m\}$
$\forall P:\Pi, A: \Phi.$	$P \models \square A$ iff $\forall P': \Pi. P \downarrow^* P' \Rightarrow P' \models A$
$\forall P:\Pi, A: \Phi.$	$P \models \Box A$ iff $\forall P'': \Pi. P \rightarrow^* P' \Rightarrow P' \models A$
$\forall P:\Pi, A: \Phi.$	$P \models A@$ iff $\forall n: \Lambda. P \models A@n$
$\forall P:\Pi, A: \Phi.$	$P \models \triangleright A$ iff $\forall P': \Pi. P P' \models A$
$\forall P:\Pi, A, B: \Phi.$	$P \models \triangleright(A \Rightarrow B)$ iff $\forall P': \Pi. P P \models A \Rightarrow P P \models B$ (cf. $P \triangleright B$ )

**18. Please amend the paragraph at page 23, lines 23-24 as follows:**

A 18 • A process  $P$  satisfies the  $A \sqcup B$  formula if for every decomposition of  $P$  into processes  $P'$  and  $P''$  such that  $P \equiv P'|P''$ , either  $P'$  satisfies  $A$  or  $P''$  satisfies  $B$ .

**19. Please amend the paragraph at page 24, lines 13-14, as follows:**

A 19 • A process  $P$  satisfies the formula  $\triangleright A$  if for every process (i.e., for every context) the combination of  $P$  and with that process satisfies  $A$ .

**20. Please amend the paragraph at page 24, lines 15-18, as follows:**

S 20 • If process  $P$  satisfies the formula  $A \triangleright B$ , it means that in every context that satisfies  $A$ , the combination (of  $P$  and the context) satisfies  $B$ . Instead, if process  $P$  satisfies the formula  $\triangleright(A \Rightarrow B)$ , it means that in every context, if the combination satisfies  $A$  then the combination satisfies  $B$ .

**21. Please amend the paragraph at page 24, line 22, as follows:**

A 21  $P \equiv P' \Rightarrow (P \models A \Rightarrow P' \models A)$

**22. Please amend the paragraph at page 24 line 27- page 25, line 33, as follows:**

A list of examples of the satisfaction relations is now provided. These examples should appear intuitively true from the definitions.

Location

$$n[] \models n[T]$$

$$n[] | 0 \models n[T], \text{ because } n[] | 0 \equiv n[]$$

$$n[m[]] \models n[m[T]]$$

$$\neg 0 \models n[T]$$

$$\neg n[] \models m[T], \text{ if } n \neq m$$

Composition

$$n[] | m[] \models n[T] | m[T].$$

$$n[] | m[] \models m[T] | n[T], \text{ because } n[] | m[] \equiv m[] | n[]$$

$$n[] | P \models n[T] | T$$

$$n[] \models n[T] | T, \text{ because } n[] \equiv n[] | 0$$

$$!n[] \models n[T] | T, \text{ because } !n[] \equiv n[] | !n[]$$

$$\neg n[] \models n[T] | n[T]$$

$$\neg n[] | n[] \models n[T]$$

$$\neg !n[] \models n[T]$$

$$\neg n[] | \text{open } m \models n[T]$$

Quantification

$$n[] \models \exists m. m[T] \text{ iff } \exists p. n[] \models p[T] \text{ iff } n[] \models n[T] \text{ iff true}$$

$$n[m[]] \models \exists n. n[n[T]] \text{ iff } \exists p. n[m[]] \models p[p[T]] \text{ iff false}$$

$$0 \models \forall n. \neg n[T]$$

Spatial Modality

$$n[m[]] \models \diamond m[T]$$

$$\neg n[m[] | m[]] \models \diamond m[T]$$

Temporal Modality

$$n[m[]] | \text{open } n \models \diamond m[T]$$

$$n[n[]] | \text{open } n \models \square(n[T] | T)$$

Location Adjunct

$$n[] \models m[n[T]] @ m$$

$n[out\ m] \models (\Diamond n[T]) @ m$

Composition Adjunct

$n[] \models m[T] \triangleright (n[T] \mid m[T])$

$\text{open } n. m[] \models (\Box n[T]) \triangleright (\Diamond m[T])$

Presence

$an\ n \triangleq n[T] \mid T$  (there is now an  $n$  here)

$no\ n \triangleq \neg an\ n$  (there is now no  $n$  here)

$one\ n \triangleq n[T] \mid no\ n$  (there is now exactly one  $n$  here)

$unique\ n \triangleq n[\Box no\ n] \mid \Box no\ n$  (there is now exactly one  $n$ , and it is here)

$!(n[T] \Rightarrow n[A])$  (every  $n$  here satisfies A)

23. Please amend the table at page 24, lines 40-42, as follows:

$vld\ A \triangleq \forall P:\Pi. P \models A$  A is valid

$sat\ A \triangleq \exists P:\Pi. P \models A$  A is satisfiable

24. Please amend the paragraph at page 26 lines 14-16, as follows:

Sequents:

$A \vdash B \triangleq vld(A \Rightarrow B)$

25. Please amend the paragraph at page 26 lines 17-22, as follows:

Rules:

$A_1 \vdash B_1; \dots; A_n \vdash B_n / A \vdash B \triangleq$

$A_1 \vdash B_1 \wedge \dots \wedge A_n \vdash B_n \Rightarrow A \vdash B \quad (n \geq 0)$

$A_1 \vdash B_1 // A_2 \vdash B_2 \triangleq$

$A_1 \vdash B_1 / A_2 \vdash B_2 \wedge A_2 \vdash B_2 / A_1 \vdash B_1$

26. Please amend the paragraph at page 26, lines 27-33, as follows:

The following is a non-standard presentation of the sequent calculus, where each sequent has exactly one assumption and one conclusion:  $A \vdash B$ . This presentation is adopted because the logical connectives introduced later do not preserve the shape of multiple-assumption multiple-conclusion sequents. Moreover, in this presentation the rules of propositional logic become extremely symmetrical. Propositional logic is summarized in the following table.

27. Please amend the table between page 26, line 33 and page 27, line 2, as follows:

(A-L)	$A \wedge (C \wedge D) \vdash B // (A \wedge C) \wedge D \vdash B$
(A-R)	$A \vdash (C \vee D) \vee B // A \vdash (C \vee D) \vee B$
(X-L)	$A \wedge C \vdash B / C \wedge A \vdash B$
(X-R)	$A \vdash C \vee B / A \vdash C \vee B$
(C-L)	$A \wedge A \vdash B / A \vdash B$
(C-R)	$A \vdash B \vee B / A \vdash B$
(W-L)	$A \vdash B / A \wedge C \vdash B$
(W-R)	$A \vdash B / A \vdash C \vee B$
(Id)	$/ A \vdash A$
(Cut)	$A \vdash C \vee B; A' \wedge C \vdash B' / A \wedge A' \vdash B \vee B'$
(T)	$A \wedge T \vdash B / A \vdash B$
(F)	$A \vdash F \vee B / A \vdash B$
(¬-L)	$A \vdash C \vee B / A \wedge \neg C \vdash B$
(¬-R)	$A \wedge C \vdash B / A \vdash \neg C \vee B$
(∧)	$A \vdash B; A' \vdash B' / A \wedge A' \vdash B \wedge B'$
(∨)	$A \vdash B; A' \vdash B' / A \vee A' \vdash B \vee B'$

**28. Please amend the paragraph at page 27, lines 6-11, as follows:**

For predicate logic the syntax of formulas (but not of processes) is enriched with variables ranging over names. These variables are indicated by letters  $x, y, z$ . Quantifiers bind variables, not names. Then, if  $fv(A) = \{x_1, \dots, x_k\}$  are the free variables of  $A$  and  $\varphi \in fv(A) \rightarrow A$  is a substitution of variables for names,  $A_\varphi$  for  $A \{x_1 \leftarrow \varphi(x_1), \dots, x_k \leftarrow \varphi(x_k)\}$  is written, and the following is defined:

$$vld A \triangleq \forall P: \prod. P \models A_\varphi$$

**29. Please amend the table at page 27, between lines 11 and 15, as follows:**

( $\forall$ -L)	$A \{x \leftarrow m\} \vdash B / \forall x. A \vdash B$	
( $\forall$ -R)	$A \vdash B / A \vdash \forall x. B$	Where $x \notin fv(A)$
( $\exists$ -L)	$A \vdash B / \exists x. A \vdash B$	Where $x \notin fv(B)$
( $\exists$ -R)	$A \vdash B \{x \leftarrow m\} / A \vdash \exists x. B$	

**30. Please amend the table at page 28, between lines 7 and 10, as follows:**

( $\Diamond$ )	$/ T \vdash \Diamond A \Leftrightarrow \neg \Box \neg A$	( $\Box$ )	$/ T \vdash \Box A \Leftrightarrow \neg \Diamond \neg A$
( $\Box K$ )	$/ T \vdash \Box(A \Rightarrow B) \Rightarrow (\Box A \Rightarrow B)$	( $\Box K$ )	$/ T \vdash \Box(A \Rightarrow B) \Rightarrow (\Box A \Rightarrow \Box B)$
( $\Box T$ )	$/ T \vdash \Box A \Rightarrow A$	( $\Box T$ )	$/ T \vdash \Box A \Rightarrow A$
( $\Box 4$ )	$/ T \vdash \Box A \Rightarrow \Box \Box A$	( $\Box 4$ )	$/ T \vdash \Box A \Rightarrow \Box \Box A$
( $\Box M$ )	$A \vdash B / \Box A \vdash \Box B$	( $\Box M$ )	$A \vdash B / \Box A \vdash \Box B$
( $\Box \wedge$ )	$\Box(A \wedge C) \vdash B // \Box A \wedge \Box C \vdash B$	( $\Box \wedge$ )	$\Box(A \wedge C) \vdash B // \Box A \wedge \Box C \vdash B$
( $\Box \vee$ )	$A \vdash \Box(\Box \vee B) // A \vdash \Box(\Box \vee \Box B)$	( $\Box \vee$ )	$A \vdash \Box(\Box \vee B) // A \vdash \Box(\Box \vee \Box B)$

**31. Please amend the paragraph at page 28, line 17- page 29, line 10, as follows:**

Finally, location properties, location rules, composition properties, and composition rules are listed.

**Location Properties**

- (1)  $vld(n[A \wedge B] \Leftrightarrow n[A] \wedge n[B])$
- (2)  $vld(n[A \vee B] \Leftrightarrow n[A] \vee n[B])$

**Location Rules**

( $n[]$ )	$A \vdash B // n[A] \vdash n[B]$
( $n[] \wedge$ )	$n[A \wedge C] \vdash B // n[A] \wedge n[C] \vdash B$
( $n[] \wedge$ )	$A \vdash n[(\wedge B)] // A \vdash n[C] \wedge n[B]$

**Composition Properties**

*b1*

- (1)  $vld(A | B \Rightarrow B | A)$
- (2)  $vld(A | (B | C) \Leftrightarrow (A | B) | C)$
- (3)  $vld((A \wedge B) | C \Rightarrow A | C \wedge B | C)$
- (4)  $vld((A \vee B) | C \Rightarrow A | C \vee B | C)$

**Composition Rules**

( )	$A' \vdash B'; A'' \vdash B'' / A'   A'' \vdash B'   B''$
(  $\wedge$ )	$(A \wedge B)   C \vdash D / A   C \wedge B   C \vdash D$
(  $\vee$ )	$A \vdash (B \vee C)   D / A \vdash B   D \vee C   D$
(  [? ]   )	$/ A'   A'' \wedge B'    B'' \vdash A'   B'' \vee B'   A''$
(  $\neg$ )	$/ \neg(A'   A'') \wedge \neg(B'   B'') \vdash \neg(B'   A'') \vee (\neg A'   \neg B'')$
(  -E)	$A \vdash B'   B''; A' \wedge (B'   C'') \vdash D; A'' \wedge (C'   B'') \vdash D$ $/ (A \wedge (A' \wedge A'')) \wedge (C'    C'') \vdash D$

**32. Please amend the paragraph at page 29, lines 14-30, as follows:**

The following propositions and corollaries relate to location adjunct rules, and composition adjunct rules. The first proposition states that  $A@n$  and  $n[A]$  are adjuncts.

Proposition: Location Adjunct Rules

$$(n[] @) \quad n[A] \vdash B // A \vdash B@n$$

Corollaries

- (1)  $vld\ n[A@n] \Rightarrow A$
- (2)  $vld\ A \Rightarrow n[A]@n$

Proposition: Composition Adjunct Rules

$$(\triangleright) \quad A | C \vdash B // A \vdash (\triangleright B)$$

Corollaries

- (1)  $vld\ A \triangleright B | B \Rightarrow B$
- (2)  $vld\ A \Rightarrow B \triangleright (A B)$
- (3)  $vld\ A \triangleright B | B \triangleright C \Rightarrow A \triangleright C$

**33. Please amend the paragraph at page 29, lines 34-39, as follows:**

In this sub-section, validity and satisfiability are reflected into the logic, by means of the  $\triangleright$

$\triangleright$  operator:

$$\begin{array}{lll} Vld\ A & \triangleq & (\neg A) \triangleright F \\ Sat\ A & \triangleq & \neg(A \triangleright F) \end{array}$$

**34. Please amend the paragraph at page 30, lines 1-22, as follows:**

From this validity and satisfiability, two propositions and one lemma are described:

Proposition:  $Vld$  and  $Sat$

- (1)  $vld \ Vld A \Leftrightarrow vld A$
- (2)  $vld \ Sat A \Leftrightarrow sat A$

Lemma:  $Vld, Sat$  Properties

- (1)  $vld (Vld(A \wedge B)) \Leftrightarrow VldA \wedge VldB$
- (2)  $vld (Vld(A \vee B)) \Leftrightarrow VldA \vee VldB$

Proposition:  $Vld, Sat$  is Modal S5

$(Sat)$	$/ T \vdash Sat A \Leftrightarrow \neg Vld \neg A$
$(Vld K)$	$/ T \vdash Vld(A \Rightarrow B) \Rightarrow ((VldA) \Rightarrow (VldB))$
$(Vld T)$	$/ T \vdash (VldA) \Rightarrow A$
$(Vld 5)$	$/ T \vdash (Sat A) \Rightarrow (Vld Sat A)$
$(Vld M)$	$A \vdash B / VldA \vdash VldB$
$(Vld \wedge)$	$Vld(A \wedge C) \vdash B // VldA \wedge VldC \vdash B$
$(Vld \vee)$	$A \vdash Vld(C \vee B) // A \vdash VldC \vee VldB$

**35. Please amend the paragraph at page 30, line 30, as follows:**

$$m = n \triangleq Vld(an m \triangleright an n)$$

**36. Please amend the paragraph at page 31, lines 2-7, as follows:**

In this section of the detailed description, examples of mobile computing environments in conjunction with the modal logic of the preceding section are presented. Specifically, four separate situations are shown in the diagram of FIG. 4, and an additional situation is shown in the diagram of FIG. 5. Those of ordinary skill within the art can appreciate that the situations of FIGs. 4 and 5 are examples for illustrative purposes only, and do not represent a limitation on the invention.

37. Please amend the paragraph at page 31, lines 8-18, as follows:

Referring first to FIG. 4, four situations are presented, situations 600, 602, 604 and 606. In situation 600, a container  $n$  includes a process  $Q$ , and includes a policy telling the container how to behave. Specifically, the policy is  $in\ m.P$ , which instructs the container  $n$  including the process  $Q$  to move into the container  $m$  already having the policy  $R$  therein, as shown in situation 600. In situation 602, a container  $n$  includes a process  $Q$ , and the policy telling the container how to behave is  $out\ m.P$ , which instructs the container  $n$  including the process  $Q$  to move out of the container  $m$  also having the policy  $R$  therein, as shown. In situation 604, the policy or instruction  $open\ n.P$  is executed on the container  $n$  having the process  $Q$ , such that  $Q$  exits the container  $n$  as a result. Finally, in situation 606, a replicated instruction is executed on the process  $P$ , such that an additional process  $P$  is made (that is, process  $P$  is copied).

38. Please amend the paragraph at page 31, line 19-23, as follows:

Referring next to FIG. 5, a communication operation referred to as a note is shown in the situation 700. The note can reside within a container. The capabilities that can be held by the note include names, such as  $n$ , as well as action capabilities, such as  $in\ n$ ,  $out\ n$ ,  $open\ n$ , or a path, such as  $C.C'$ , as has been described in the modal logic section of the detailed description.

39. Please amend the paragraph at page 32, line 22- page 33, line 8, as follows:

302, 304, 306, 308, 310, 312, and 314 implement the analysis of the process against a formula, using a predetermined modal logic based on ambient calculus, according to one embodiment of the invention. The formula against which the process is to be analyzed can be a policy, such as a security policy or a mobility policy, such that the policy is described using the predetermined modal logic, such as has been described in the preceding sections of the detailed description. In one embodiment, the process is analyzed in a recursive manner. The analysis of 302, 304, 306, 308, 310, 312, and 314 can be summarized as a theorem, specifically, for all replication-free process  $P$  and  $\triangleright$ -free closed formulas  $\mathbb{A}$ ,  $P \models \mathbb{A}$  if and only if  $Check(P, \mathbb{A})$ , where  $Check()$  is the analysis of 302, 304, 306, 308, 310, 312, and 314.

**40. Please amend the paragraph at page 34, lines 1-4 as follows:**

In 306 specifically, the process is partitioned to determine whether each component of the process satisfies the formula, or policy. If any component fails against the policy, then the process itself fails. The check of 306 only applies if the formula is a composition A|B. This check can be expressed as:

---